

## KOMPRESI PADA CITRA DIGITAL MENGUNAKAN ALGORITMA RUN LENGTH ENCODING

Winda Kusdianti<sup>1</sup>, Anindita Septiarini<sup>2</sup>

Program Studi Ilmu Komputer, FMIPA Universitas Mulawarman

[winda.kusdianti@gmail.com](mailto:winda.kusdianti@gmail.com)<sup>1</sup>, [anindita.septiarini@gmail.com](mailto:anindita.septiarini@gmail.com)<sup>2</sup>

**Abstrak.** Kompresi citra bertujuan untuk mengurangi ukuran *file* untuk merepresentasikan suatu citra digital sehingga dapat menghemat memori penyimpanan dan mempercepat proses pertukaran data. Pada penelitian ini *file* citra dikompres menggunakan algoritma *Run Length Encoding* (RLE). Algoritma ini bekerja dengan mengelompokkan nilai derajat keabuan yang sama dari sebuah citra kemudian menyimpannya dalam format PNG sehingga tidak terjadi penurunan kualitas warna (kompresi) serta dapat mengembalikan nilai derajat keabuan yang sama dari citra tersebut (dekompresi). Hasil pengujian sistem menunjukkan proses kompresi menghasilkan dua buah *file* PNG yaitu citra *pixel* dan citra populasi dengan ukuran *file* citra terkompresi yang dihasilkan beragam, pada format citra BMP memiliki rasio sebesar -0,08 % hingga 97 % dan pada format citra TIFF sebesar 28% hingga 74 % tergantung pada banyaknya nilai derajat keabuan yang sama dan berurutan pada masing-masing citra tersebut. Semakin homogen warna suatu citra digital, rasio kompresi yang dihasilkan akan semakin besar dan semakin sedikit waktu yang dibutuhkan untuk melakukan proses kompresi.

**Kata Kunci :** *Kompresi, Dekompresi, Citra Digital, Run Length Encoding (RLE).*

Pada era digital seperti sekarang, perkembangan teknologi kamera semakin canggih sehingga dapat menghasilkan citra dengan resolusi yang tinggi. Hal ini tentu saja membutuhkan tempat penyimpanan memori yang besar pula. Besarnya ukuran *file* citra juga dapat berpengaruh pada lamanya waktu yang dibutuhkan untuk melakukan suatu pengiriman citra (transmisi). Salah satu solusi untuk masalah-masalah tersebut adalah melakukan kompresi citra.

Kompresi citra bertujuan untuk mengurangi ukuran citra digital sebelum menyimpan atau memindahkan ke dalam media penyimpanan. Salah satu algoritma kompresi yaitu RLE yang termasuk ke dalam tipe kompresi *lossless*. Algoritma ini cocok untuk mengkompresi *file* yang mengandung informasi penting yang tidak boleh rusak akibat kompresi. Algoritma RLE bekerja dengan cara mengelompokkan *pixel-pixel* yang memiliki warna yang sama secara berurutan sehingga dapat mengurangi duplikasi warna pada citra digital.

Citra digital yang digunakan yaitu format BMP dan TIFF dengan model warna *grayscale* dan RGB hingga kedalaman 24 bit. Proses kompresi menghasilkan dua buah citra yaitu citra *pixel* dan citra populasi dengan

format PNG sedangkan proses dekompresi menghasilkan citra dengan format PNG, BMP atau TIFF. Pengujian sistem didasarkan pada hasil rasio kompresi, lamanya waktu yang dibutuhkan untuk melakukan proses kompresi dan perbandingan ukuran *file* hasil dekompresi berupa format PNG dengan format asli citra tersebut.

### METODOLOGI

Kompresi data sangat populer karena dua alasan yaitu [3]:

1. Orang-orang lebih suka mengumpulkan data. Tidak peduli seberapa besar media penyimpanan yang dimilikinya. Akan tetapi cepat atau lambat akan terjadi *overflow*.
2. Orang-orang kurang suka menunggu dalam waktu yang lama untuk memindahkan data. Misalnya ketika duduk di depan komputer untuk menunggu halaman *website* terbuka atau men-*download* sebuah *file*.

Secara garis besar klasifikasi metode kompresi data dapat dibedakan menjadi dua [1]:

1. Metode *Lossless*

Pada teknik ini tidak ada kehilangan data atau informasi. Jika data dikompres secara *Lossless*, data asli dapat direkonstruksi kembali sama persis dari data yang telah dikompresi, Contoh algoritma yang termasuk metode ini antara lain *Shannon-Fano Coding*, *Run Length Encoding* dan *Arithmetic Coding*.

## 2. Metode *Lossy*

Pada teknik ini akan terjadi kehilangan sebagian informasi atau data. Data yang telah dikompresi dengan teknik ini secara umum tidak bisa direkonstruksi sama persis dari data aslinya. Contoh algoritma yang termasuk metode ini antara lain *Transform Coding*, *Wavelet* dan *Fraktal*.

RLE termasuk kedalam tipe kompresi *Lossless* sederhana. RLE didasarkan pada pengamatan bahwa jika memilih *pixel* dalam gambar secara acak, ada kesempatan bahwa tetangga akan memiliki warna yang sama. Sehingga saat kompresi, sistem dapat mencari baris demi baris bitmap warna yang sama (*scanning*).

Pada dasarnya citra digital dibentuk dengan cara pendefinisian nomor warna disetiap *pixel* yang terdapat didalam citra digital itu sendiri. Pendefinisian warna sendiri merupakan hasil perkalian lebar dengan tinggi citra. Warna yang digunakan kemungkinan besar banyak mengalami pengulangan untuk lokasi yang berbeda sehingga dapat menambah ukuran *file* citra tersebut. Ukuran rasio kompresi tergantung pada kompleksitas gambar itu sendiri, semakin detail gambar maka hasil kompresi menjadi kurang optimal [2].

Sebagaimana algoritma-algoritma lainnya, algoritma RLE juga memiliki kelebihan dan kekurangan dalam penggunaannya. Algoritma ini akan bekerja dengan baik untuk citra yang memiliki data nilai informasi warna yang sama (homogen). Namun ketika algoritma RLE dihadapkan pada kondisi data nilai informasi warna yang memiliki populasi yang sangat beragam beragam, maka hasil kompresi dari algoritma ini akan lebih besar dari *file* aslinya. Pernyataan ini dapat dibuktikan pada simulasi berikut.

- Data pertama:

PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPRRRRRRRRRR

Dikompres menggunakan Algoritma RLE:  
P13Q7P3R8

Rasio kompresi:  $\frac{(31-8)}{31} \times 100\% = 74.2\%$

- Data kedua:

PPPQQQRRRRQQSSPPSQQRQPRRSSSS  
PSS

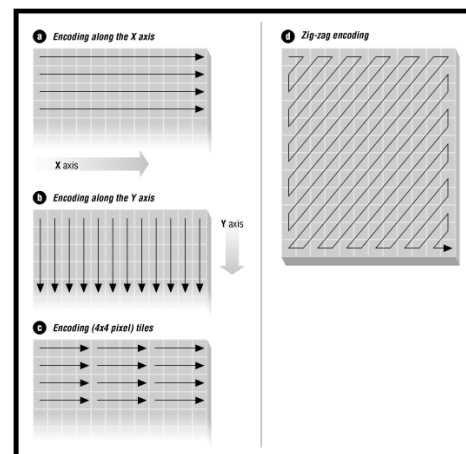
Dikompres menggunakan Algoritma RLE:  
P3Q4R3Q3S2P2S1Q2R1Q1P1R2S3P1S2

Rasio kompresi:  $\frac{(31-30)}{31} \times 100\% = 3.2\%$

Dari hasil simulasi menggunakan jumlah data yang sama, didapatkan hasil yang berbeda pula. Data pertama menghasilkan rasio kompresi lebih besar jika dibandingkan dengan data kedua. Hal ini menandakan bahwa rasio kompresi RLE dipengaruhi oleh kombinasi populasi data.

Algoritma RLE dapat dibedakan menjadi 4 varian dilihat dari proses *scanning* data citra.

Varian RLE berdasarkan *scanning pixel* citra dapat dilihat pada gambar 1 [2].



Gambar 1. Varian RLE

Dari gambar 1 terdapat empat tipe varian RLE yaitu gambar 1 sub a *scanning* citra secara *vertical* perbaris (X axis), gambar 1 sub b *scanning* citra secara *horizontal* perkolom (Y axis), gambar 1 sub c *scanning* citra dikodekan berkelompok menjadi 4 x 4 *pixel* dan gambar 1 sub d *scanning* citra secara zig-zag. Varian RLE seperti gambar sub d sangat jarang digunakan dan kemungkinan hanya digunakan dalam aplikasi-aplikasi khusus.

Rasio kompresi data adalah ukuran persentase data yang berhasil dikompres. Secara matematis persamaan rasio kompresi data ditulis sebagai berikut [3]:

$$R = \frac{M0 - M1}{M0} \times 100\% \dots \dots \dots (1)$$

Dari persamaan 1, R menunjukkan rasio kompresi berupa persentase kemudian M0 adalah ukuran *file* asli sebelum dikompres dan M1 adalah ukuran *file* terkompresi.

Pada dasarnya RLE dapat digunakan untuk mengkompresi semua jenis data, tetapi isi data akan mempengaruhi rasio kompresi yang dicapai oleh RLE tersebut. Dalam penelitian ini format citra kompresi RLE yang didukung adalah format bitmap seperti TIFF dan BMP.

Langkah-langkah dalam kompresi citra digital menggunakan RLE adalah:

1. Citra ditampilkan dalam bentuk matriks *pixel*.
2. Berdasarkan tampilan matriks, kemudian data citra akan disusun kedalam bentuk baris satu larik dimana data *pixel* citra dan data populasi citra akan dibedakan menggunakan metode RLE dengan cara menghitung *run-length* untuk setiap derajat keabuan yang berurutan secara *vertikal* (perkolom).
3. Data citra berupa citra *pixel* dan citra populasi satu larik akan diubah menjadi citra berukuran persegi (lebar dan tinggi sama) dengan terlebih dahulu mencari nilai resolusi optimum dari citra tersebut dengan persamaan 2 dan 3.  

$$\text{Syarat } \max i = [(i-1)^2 < w \leq (i+1)^2] \quad (2)$$
Maka nilai resolusi optimum yaitu:  

$$r = \max i + 1 \quad (3)$$

Resolusi optimum didapatkan dengan mengasumsikan nilai resolusi optimum (r), nilai acuan *pixel* (w), nilai kuadrat bilangan (i). Nilai kuadrat bilangan berupa angka dari satu (1) hingga (n). Nilai n sendiri tidak terhingga sampai mendapatkan nilai yang sesuai.

## HASIL DAN PEMBAHASAN








Sistem ini memiliki dua fungsi, antara lain:

1. Mengkompresi citra digital dengan algoritma RLE yang telah ditetapkan untuk pengkodeannya dengan *scanning pixel* citra secara vertikal.
2. Mendekompresi citra digital yang telah dikompres sebelumnya dengan ketentuan pengkodean RLE sehingga citra dapat dikembalikan seperti citra aslinya.

Pengujian kompresi dan dekompresi dilakukan untuk menguji tingkat keberhasilan, apakah suatu citra dapat dikompres menggunakan algoritma RLE dengan menjadikan rasio dan waktu kompresinya

sebagai tolak ukur serta citra dapat didekompresi sehingga informasi atau kualitas citra dapat terbaca kembali. Citra yang diuji memiliki format BMP dan TIFF sedangkan pada bagian dekompresi juga akan diuji hasil proses dekompresi dari citra terkompresi dengan format PNG dan format asli citra tersebut untuk dibandingkan mana yang menghasilkan kualitas yang lebih baik. Beberapa contoh citra pengujian sistem dapat dilihat pada tabel 1.

Tabel 1 Spesifikasi Citra Pengujian




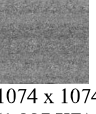
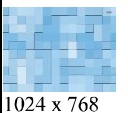
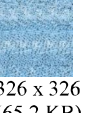

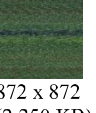

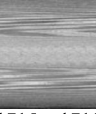



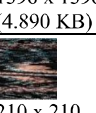
No	Nama File Citra	Tipe	Ukuran (KB)	Keterangan
1	Disket	BMP	2.304,05	 1024 x 768
2	California Kentucky Map	BMP	8.214,08	 3180 x 2645
3	Blue square	BMP	2.304,05	 1024 x 768
4	Merak	BMP	2.304,05	 1024 x 768
5	Medical Rontgen	TIFF	5.001,88	 2043 x 2494
6	Baliho caleg	TIFF	19.405,01	 2000 x 1500
7	Steve quote	TIFF	133,13	 500 x 500

## Pengujian Pada Proses Kompresi

Setiap citra yang diuji menggunakan algoritma RLE menghasilkan dua buah citra yaitu citra *pixel* untuk menampung *pixel* citra asli dan citra populasi untuk menampung jumlah *pixel* pada citra dengan resolusi terbaru berordo sama, namun citra *pixel* saja yang ditampilkan karena citra populasi hanya berupa citra kosong untuk menampung jumlah *pixel*.

Hasil lengkap pengujian pada citra kompresi ditunjukkan pada tabel 2.

Tabel 2 Hasil Pengujian Proses Kompresi Citra

	Citra Asli	Citra Terkompresi (.png)	Rasio (%)	Waktu Proses (Detik)
1	 1024 x 768	 761 x 761 (1.114 KB)	51,69	12,38
2	 3180 x 2645	 1074 x 1074 (1.337 KB)	83,74	59,90
3	 1024 x 768	 326 x 326 (65.2 KB)	97,17	3,43
4	 1024 x 768	 872 x 872 (2.250 KB)	-0,08	15,7
5	 2043 x 2494	 1715 x 1715 (3.386 KB)	32,31	101,54
6	 2000 x 1500	 1396 x 1396 (4.890 KB)	74,17	55,83
7	 500 x 500	 210 x 210 (94.7 KB)	28,79	1,564

### Pengujian Pada Proses Dekompresi

Pengujian pada proses dekomposisi merupakan pengujian untuk melihat apakah citra digital yang telah dikompresi dapat dikembalikan seperti semula dan memiliki kualitas yang sama dengan citra aslinya. Pada pengujian ini, citra akan didekomposisi kedalam dua format yaitu PNG sebagai format bawaan proses dekomposisi dan format asli dari citra tersebut (BMP atau TIFF) sehingga dapat dilihat perbandingan ukuran dan kualitas agar mendapatkan hasil dekomposisi terbaik.

Hasil lengkap pengujian pada citra dekomposisi ditunjukkan pada tabel 3.

### SIMPULAN

Dari hasil penelitian yang telah dilakukan, maka dapat disimpulkan bahwa kompresi citra digital menggunakan algoritma RLE dapat memperkecil ukuran *file* citra sehingga akan menghemat memori penyimpanan serta dapat mempercepat proses pengiriman. Proses kompresi menghasilkan dua buah *file* citra yaitu citra *pixel* dan citra populasi dengan format PNG. Citra yang sudah dikompresi menggunakan algoritma RLE dapat dikembalikan seperti semula (dekomposisi) sehingga tidak ditemukan adanya perubahan kualitas citra. Jika citra didekomposisi dengan format selain PNG maka ukuran citra rata-rata akan cenderung lebih besar dari citra asli. Pada format citra BMP memiliki rasio sebesar -0,08 % hingga 97 % dengan waktu antara 3 hingga 60 detik sedangkan pada format citra TIFF sebesar 28% hingga 74 % dengan waktu antar 1,5 hingga 102 detik jadi semakin besar dan beragam *pixel* warna maka semakin besar pula rasio kompresinya dan waktu yang diperlukan untuk proses kompresi. Didapat pula bahwa jika hasil citra kompresi diubah ukurannya, maka citra tersebut tidak dapat didekomposisi kembali seperti citra asli karena adanya perubahan jumlah *pixel*.

### DAFTAR PUSTAKA

- [1] Fauzi, R. 2009. "Modifikasi Algoritma Arithmetic Coding Dalam Mengatasi Kelemahan Pada Kompresi File Teks". *Skripsi Ilmu Komputer*, Universitas Sumatera Utara.
- [2] Salomon, D. 2004. *Data Compression The Complete Reference* 3th Edition. London: Springer-Verlag.
- [3] Salomon, D. 2007. *Data Compression The Complete Reference* 4th Edition. London: Springer-Verlag.